IN THE SPECIFICATION:

Please **amend the paragraph beginning on page 1, line 27**, as follows:

Referring to Fig. 1 herein, there is illustrated schematically a basic architecture of a prior art cluster of computer entities, in which all data storage 100 is centralized, and a plurality of processors 101 - 109 **[[**link**]]** linked together by **[[**a**]]** high-speed interface 110 operate collectively to provide data processing power to a single application, and accessing the centralized data storage device 100. This arrangement is highly scalable, and more data processing nodes and more data storage capacity can be added.

Please **amend the paragraph beginning on page 6, line 26**, as follows:

Fig. 1, ~~illustrates schematically~~ as described, is a schematic illustration of a prior art cluster arrangement of conventional computer entities, having user consoles allowing operator access at each of a plurality of data processing nodes;

Please **amend the paragraph beginning on page 6, line 30**, and **ending on page 7, line 2**, as follows:

Fig. 2 ~~illustrates schematically~~ is a schematic illustration of a plurality of headless computer entities connected by a local area

network, and having a single administrative console computer entity having a user console with video monitor, keyboard and tactile pointing device according to a specific implementation of the present invention;

Please **amend the paragraph beginning on page 11, line 11**, as follows:

Referring to Fig. 4 herein, there is illustrated schematically physical and logical components of a computer entity 400. The computer entity comprises a communications interface 401, for example a local area network card such as an Ethernet card; a data processor 402, for example an Intel® Pentium or similar Processor; a memory 403, a data storage device 404, in the best mode herein an array of individual disk drives in a RAID (redundant array of inexpensive disks) configuration; an operating system 405, for example the known Windows 2000®, Windows95, Windows98, Unix, or Linux operating systems or the like; a display 406, such as an LCD display; a web administration interface 407 by means of which information describing the status of the computer entity can be communicated to a remote display; an aggregation service module 408 in the form of an application**[[,]]** for managing the data storage device within a group environment; and one or a plurality of applications programs 409, capable of being synchronised with other applications on other group member computer entities.

Please **amend the paragraph beginning on page 11, line 27,** and **ending on page 12, line 14,** as follows:

Referring to Fig. 5 herein, there is illustrated schematically a partition format of a headless computer entity, upon which one or more operating system(s) are stored. Data storage device **[[**400**]]** 500 is partitioned into a logical data storage area which is divided into a plurality of partitions and sub-partitions according to the architecture shown. A main division into a primary partition 501 and a secondary partition 502 is made. Within the primary partition are a plurality of sub partitions including a primary operating system system partition 503 (POSSP), containing a primary operating system of the computer entity; an emergency operating system partition 504 (EOSSP) containing an emergency operating system under which the computer entity operates under conditions where the primary operating system is inactive or is deactivated; an OEM partition 505; a primary operating system boot partition 506 (POSBP), from which the primary operating system is booted or rebooted; an emergency operating system boot partition 507 (EOSBP), from which the emergency operating system is booted; a primary data partition 508 (PDP) containing an SQL database 509, and a plurality of binary large objects 510, (BLOBs); a user settings archive partition 511 (USAP); a reserved space partition 512 (RSP) typically having a capacity of the order of 4 gigabytes or more; and an operating system back up area 513 (OSBA)

containing a back up copy of the primary operating system files 514. The secondary data partition 502 comprises a plurality of binary large objects 515.

Please **amend the paragraph beginning on page 12, line 29**, and **ending on page 13, line 4**, as follows:

Each slave headless computer entity, 601, 602 is loaded with **[[a]]** the same slave aggregation service application 608, 609 and **[[a]]** the same slave user application 610, 611. Modifications to the configuration of the master user application 606 of the master computer entity are automatically propagated by the master aggregation service application 607 to all the slave user applications 610, 611 on the slave computer entities. The master aggregation service application on the master headless computer entity 600 automatically synchronizes all of its settings to all of the slave computer entities 601, 602.

Please **amend the paragraph beginning on page 13, line 18**, as follows:

Referring to Fig. 7 herein, there is illustrated logically an aggregation service provided by an aggregation service application 700, along with modes of usage of that service by one or more agents 701, data management application 702, and by a user via web administration interface 703. In each case, the aggregation service

responds via a set of Application Procedure Instruction (API) calls, which ~~interfaces~~ interface with the operating system on the master headless computer entity. Operations are then propagated from the operating system on the master computer entity, to the operating systems on each of the slave headless computer entities, which, via the slave aggregation service applications 608, 609, make changes to the relevant applications on each of the slave computer entities.

Please **amend the paragraph beginning on page 13, line 29**, and **ending on page 14, line 31**, as follows:

Referring to Fig. 8 herein, there is illustrated schematically process steps carried out by an aggregation service master application in conjunction with an operating system, at a master computer entity for adding a new slave computer entity to a group. In step 800, a user, from the headed console 207 selects a headless computer entity by searching the network for attached computer entities. Searching may be made via a prior art searching facility provided as part of the prior art operating system of the master computer entity, accessed through the web interface. In step 801, where the headless computer entity is to be added to an existing group, the user selects, via the conventional computer entity 207 and web interface of the master computer entity 600, an existing group to which the headless computer entity is to be added in the capacity of the slave. The master computer entity **[[**may**]]** can manage several

different groups simultaneously. Selection is achieved by selecting an existing group from a drop down menu displayed by the web administration interface 605. In step 802, the master computer entity sends configuration settings to the newly added slave computer entity, so that the slave computer entity can authenticate itself as being part of the group. Authentication by the slave computer entity comprises receiving data from the master computer entity describing which group the slave computer entity has been assigned to, and the slave computer entity storing that data within a database in the slave. In step 803, the master computer entity authenticates the new slave entity within the group listings stored at the master, by adding data describing the address of the slave entity, and describing operating system configuration settings and application configuration settings applied to the slave computer entity in a database listing stored at the master. In step 804, if addition of the slave computer entity to the group has not been successful, either because the addition to the group has not been authenticated on the slave or the master, then the aggregation service 607 API returns an error code to the web interface (step 805). The error code would typically arise due to a routine for making checks for adding the new slave computer entity to the group failing. In this case, in step 806, the master application 606 displays via the web admin interface 605 an error dialogue, readable at the headed computer

entity 207, indicating that the addition to the group has failed. Thus, steps 802-804 constitute an interlock that ensures that operating system changes resulting from adding a new slave do not break connections across the group including the new slave. However if in step 804 the slave computer entity is successfully added to the group, then in step 807 the slave computer entity is added to the list of slave entities in the group, by adding an object describing the entity to the group in step 808.

Please **amend the paragraph beginning on page 15, line 12**, as follows:

The user interface display illustrated in Fig. 9 **[[**shows**]]** includes a listing of a plurality of groups, in this case a first group *Auto Back Up 1* comprising a first group of computer entities, and a second group *Auto Back Up 2* comprising a second group of computer entities.

Please **amend the paragraph beginning on page 15, line 17**, as follows:

Within the first group *Auto Back Up 1*, objects representing individual slave computer entities appear in sub groups including a first sub group protected computers, a second sub group (users), and a third sub group (appliance maintenance).

Please **amend the paragraph beginning on page 17, line 23**, as follows:

In each group, the first appliance to use **[[to]]** or create the group is designated as the "master", and then "slave" computer entities are added to the group. The master entity in the group is used to store the group level configuration settings for the group, which the other slave computer entities synchronize themselves in order to be in the group.

Please **amend the paragraph beginning on page 17, line 29**, and **ending on page 18, line 19**, as follows:

Referring to Fig. 11 herein, there is illustrated schematically actions taken by the aggregation service master application 607 when a new computer entity is successfully added to a group (step 1100). The aggregation service master application 607 resident on the master computer entity 600 automatically synchronizes the security settings of each computer entity in the group in step 1101. This is achieved by sending a common set of security settings across the network, addressed to each slave computer entity within the group. When each slave computer entity receives those security settings, each slave computer entity self applies those security settings to itself. In step 1102, the aggregation service 607 synchronizes a set of time zone settings for the new appliance added to the group. Time zone

settings will already exist on the master computer entity 600, (and on existing slave computer entities in the group). The time zone settings are sent to the new computer entity added to the group, which then applies those time zone settings on the slave aggregation service application in that slave computer entity, bringing the time zone settings of the newly added computer entity in line with those computer entities of the rest of the group. In step 1103, any global configuration settings for a common application in the group are sent to the client application on the newly added computer entity in the group. The newly added computer entity applies those global application configuration settings to the slave user application running on that slave computer entity, bringing the settings of that slave user application, into line with the configuration settings of the server application and any other client applications within the rest of the group.

Please **amend the paragraph beginning on page 18, line 21** as follows:

Referring to Fig. 12 herein, there is illustrated schematically actions taken by the master application 606 when a computer entity group is created. The actions are taken when a new computer entity group is created, by the user ~~application~~ applications 605, 610, 611, which the group serves. The relevant commands need to be written into

the user application, in order that the user application will run on the group of headless computer entities.

Please **amend the paragraph beginning on page 19, line 1,** as follows:

In step 1200, a first type of data management application configuration setting comprising global maintenance properties, is synchronized across all computer entities in the group. The global maintenance properties ~~includes~~ include properties such as scheduled back up job throttling; and appliance maintenance job schedules. These are applied across all computer entities in the group by the aggregation service applications 607, **[[**with**]]** in response to the data being ~~input from~~ supplied to applications 607 by the master management application 606.

Please **amend the paragraph beginning on page 19, line 19,** as follows:

In step 1202, a third type of data management application configuration settings~~, are~~ is applied such that any protected computer groups and their properties are synchronized across the group. The properties synchronized to the protected computer groups includes schedule; retention; excludes; rights; limits and quotas; log critical files, and data file definitions applicable to protected computer groups. Again, this is effected by the master management

application 606 applying those properties through the aggregation service 607 which sends data describing those properties to each of the computer entities within the group, which then self apply those properties to themselves.

Please **amend the paragraph beginning on page 19, line 29,** and **ending on page 20, line 2,** as follows:

An advantage of the above implementation is that it is quick and easy to add a new computer entity into a group of computer entities. The only synchronization between computer entities required is **[[**of**]]** group level configuration settings. There is no need for a distributed database merge operation, and there is no need to merge a new computer entities file systems into a distributed network file system shared across all computer entities.

Please **amend the paragraph beginning on page 20, line 10,** as follows:

In step 1300 the *add to group* API receives a request to add a new computer entity to an existing group, the request being generated from the data management application 606, in response to the request via the web interface, input through the administration console. Before adding a new computer entity to a group, the aggregation service application 607 checks, in step 1301 whether the new slave computer entity to be added to the group has **[[**a**]]** <u>the</u> same generic,

or "NT Domain" security mode setting as the first computer entity (the master computer entity) in the group. If the slave computer entity does not have the same generic or "NT Domain" security mode setting as the master computer entity, then the request to add the new slave computer entity to the group is rejected in step 1303, and in step 1304 an error message is generated alerting the user via the web interface/or LCD that the security mode must be the same across the whole of the group. Thus, steps 1301 and 1302 constitute an interlock that ensures that operating system changes resulting from adding a new slave do not break connections across the group including the new slave. However, if the generic or "NT Domain" security mode settings between the master computer entity and slave computer entity are found to be the same in step 1301 then in step 1302, the addition of the new slave computer entity to the group proceeds.

Please **amend the paragraph beginning on page 20, line 27,** and **ending on page 21, line 10**, as follows:

Referring to Fig. 14, there is illustrated schematically processes carried out by the *add to group* API when a new computer entity is to be added to an existing group, and the existing computer entities in that group are using an NT domain security mode. In step 1400, the API receives the request to add a new computer entity to the existing group and in step 1401 the aggregation service

application checks the security mode of existing computer entities in the group, to make sure that they are NT domain. Where this is the case, then in step 1402 the aggregation service application checks whether the new computer entity is configured to be in the same domain as the other computers in the group. If not, then in step 1403 the aggregation service application rejects the request to add the new computer entities to the group, and in step 1404 displays an error dialogue box via the web interface and/or LCD, alerting an administrator that all members in the group must be in the same NT domain. If in step 1402 the new computer entity is configured to be in the same NT domain security mode as the other computers in the group, then in step 1405 the new computer entity can proceed to be added to the group.

Please **amend the paragraph beginning on page 21, line 12**, as follows:

Referring to Fig. 15 herein, there is illustrated schematically processes carried out by the aggregation service application 607 for adding a new computer entity to an existing group of computer entities. In step 1500, a request to add a new computer entity to an existing group is received from a management application 606 as herein before described. In step 1501 the aggregation service application checks if any computers in the group use DHCP configuration. Having established that all existing computers in the

group do use DHCP configuration, there is applied the basic assumption that all computers are on a same logical network, that is to say there are no routers between different computer entities in the same group (step 1502). In step 1503 it is checked whether the master computer entity is using DHCP configuration. If so, then in step 1504, it is checked whether the master computer entity can use the UDP broadcast based IP provisioning to connect to the new computer entity by name. Thus, steps 1503-1507 constitute an interlock that ensures that operating system changes resulting from adding a new slave do not break connections across the group including the new slave. If in step 1505 it is checked whether the slave computer entity uses DHCP configuration and if so, then in step 1506 it is checked that the slave computer entity can use UDP broadcast based IP provisioning to connect to the master computer entity by name. If any of these connectivity checks fail, then in step 1507 the request to add a new computer entity to the group is rejected and in step 1508 an error warning message is displayed in the web interface and/or LCD display that all members of the group must be on the same sub-net if DHCP configuration is used.